ANL-ARC-265

SAS4A/SASSYS-1 Modernization

Nuclear Engineering Division

About Argonne National Laboratory
Argonne is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC under contract DE-AC02-06CH11357. The Laboratory's main facility is outside Chicago, at 9700 South Cass Avenue, Argonne, Illinois 60439. For information about Argonne and its pioneering science and technology programs, see www.anl.gov.

DOCUMENT AVAILABILITY

Online Access: U.S. Department of Energy (DOE) reports produced after 1991 and a growing number of pre-1991 documents are available free via DOE's SciTech Connect (http://www.osti.gov/scitech/)

Reports not in digital format may be purchased by the public from the National Technical Information Service (NTIS):
U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Rd
Alexandria, VA 22312
www.ntis.gov
Phone: (800) 553-NTIS (6847) or (703) 605-6000
Fax: (703) 605-6900
Email: orders@ntis.gov

Reports not in digital format are available to DOE and DOE contractors from the Office of Scientific and Technical Information (OSTI):
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831-0062
www.osti.gov
Phone: (865) 576-8401
Fax: (865) 576-5728
Email: reports@osti.gov

Disclaimer
This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor UChicago Argonne, LLC, nor any of their employees or officers, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of document authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof, Argonne National Laboratory, or UChicago Argonne, LLC.

ANL-ARC-265

SAS4A/SASSYS-1 Modernization

prepared by

T. H. Fanning, F. E. Dunn, D. Grabaskas, T. Sumner, and J. W. Thomas
Nuclear Engineering Division, Argonne National Laboratory

September 20, 2013

ABSTRACT

SAS4A/SASSYS-1 is a software simulation tool used to perform deterministic analysis of anticipated events as well as design basis and beyond design basis accidents for advanced nuclear reactors. This report summarizes recent tasks to modernize the SAS4A/SASSYS-1 code system to improve internal data management and to update the code documentation to reflect recent code developments. The motivation for performing these updates stems from the relevance of SAS4A/SASSYS-1 to a number of U.S. Department of Energy programs as well as domestic and international collaborations.

Considerable progress has been made in modernizing the internal data management by developing data modules that organize the simulation data in a way that is consistent with modern programming practices. This eliminates most of the legacy data management practices that were essential on older, limited memory machines. Once verification tests are completed, these updates will be released as SAS4A/SASSYS-1 Version 5.1.

Updates to the previous code manual have been completed to include documentation for new modeling capabilities that have been incorporated into SAS4A/SASSYS-1 over the last several years. New modeling capabilities include a detailed sub-channel model for intra-assembly flow and temperature simulations; visualization capabilities for sub-channel results; an extended decay heat model to support advanced, actinide-bearing fuels; a coupling capability to use external computational fluid dynamics solvers to resolve flow distribution and thermal stratification effects; and treatment of axial expansion feedback from assembly duct walls. The extensive, 2100-page code manual is being published as *The SAS4A/SASSYS-1 Safety Analysis Code System, Version 5*, ANL/NE-13/13, Nuclear Engineering Division, Argonne National Laboratory, September 2013.

TABLE OF CONTENTS

LIST OF FIGURES

1. Introduction

This report documents fiscal year 2013 activities that developed and implemented modern data management practices for the SAS4A/SASSYS-1 safety analysis code system and updated the related manual to provide code documentation for significant modeling developments that had taken place over the last several years. This report satisfies the deliverable for the Level 3 milestone M3AR-13AN1701041, "SAS4A/SASSYS-1 Code Revisions"

SAS4A/SASSYS-1 is a software simulation tool used to perform deterministic analysis of anticipated events as well as design basis and beyond design basis accidents for advanced nuclear reactors. With its origin as SAS1A in the late 1960s, the SAS series of codes has been under continuous development for over forty-five years. The most significant development period was in the 1970s and 1980s to support the Fast Flux Test Facility and the Clinch River Breeder Reactor program. Considerable development continued into the 1990s in support of the Advanced Liquid Metal Reactor program, during which the modeling emphasis was on inherent safety and metallic fuel performance. In the twenty years since the termination of the ALMR program, modeling developments continued at a much slower pace.

The decline in development effort on SAS4A/SASSYS-1 ironically coincided with rapid growth in the development of modern, object-oriented programming practices that greatly facilitate code development and maintenance. This left SAS4A/SASSYS-1 in a state that reflects legacy software development practices. It is well beyond the scope of the current activities to reconstruct SAS4A/SASSYS-1 as an object-oriented code. But significant improvements in the ability to maintain and update the code can be achieved by modernizing the data-management strategies from those that were developed for resource-limited computing hardware.

One objective of this work is to modularize the data management in SAS4A/SASSYS-1 so that future developments can take advantage of modern programming practices without being constrained by the legacy data structures. All updates are implemented in a standards-compliant, portable way to maintain compatibility with Linux, Mac OS X, and Microsoft Windows platforms. A second objective of this work is to update the code documentation to include the developments that have been made during the last twenty years. Documentation of these developments were previously scattered across several conference papers, internal reports, and memos.

Modernization of the SAS4A/SASSYS-1 code system and updates to the code documentation are motivated by the relevance of the simulation capability to a number of U.S. Department of Energy programs as well as domestic and international collaborations.[1] Active programs and collaborations that currently use SAS4A/SASSYS-1 include the following:

- EBR-II IAEA Benchmark: The DOE-NE Advanced Reactor Concepts program is supporting a high-profile Coordinated Research Project with the International Atomic Energy Agency based on the Shutdown Heat Removal Tests conducted at EBR-II. Both protected (SHRT-17) and unprotected (SHRT-45R) loss-of-flow tests are part of the benchmark activity. SAS4A/SASSYS-1 models of both tests are being developed to provide results under the CRP.[2]

- ASTRID Collaboration with CEA: An implementation agreement is being developed between the U.S. DOE and the Commissariat à l'énergie atomique et aux énergies alternatives of France for cooperation in low carbon energy technologies. The purpose of the agreement is to evaluate the safety performance of the ASTRID reactor design. DOE will accomplish this using the SAS4A/SASSYS-1 safety analysis code.[3]
- CIAE Bilateral Collaboration: The DOE-NE Office of International Nuclear Energy Policy and Cooperation has established the U.S.–China Bilateral Civil Nuclear Energy Cooperative Action Plan with the China Institute of Atomic Energy. Joint activities under the action plant include model development and safety analyses of the China Experimental Fast Reactor using SAS4A/SASSYS-1.[4]
- aSMR Passive System Reliability: The DOE-NE Advanced Reactor Concepts program has conducted an evaluation of risk assessment methods that could be applied to passive safety systems.[5] One of the recommendations is to perform *dynamic* probabilistic risk assessment methods. Tentative plans for FY14 include coupling the ADAPT tool[6] to SAS4A/SASSYS-1 to perform multiple dynamic event-tree simulations for SFR passive systems.
- TerraPower TWR Reactor Concept: TerraPower, LLC has licensed the SAS4A/SASSYS-1 source code to perform safety analysis studies for their "Traveling Wave Reactor" concept. TerraPower also funds code development activities that improve the modeling capabilities of SAS4A/SASSYS-1.
- KAERI PG-SFR: The Korean Atomic Energy Research Institute has acquired a license for SAS4A/SASSYS-1 to perform safety analysis and model development for the "Prototype Generation-IV Sodium Fast Reactor". KAERI is supporting metallic fuel severe accident model developments that will be incorporated into SAS4A/SASSYS-1.
- KINS: The Korea Institute of Nuclear Safety is an independent regulatory expert organization that supports the Nuclear Safety and Security Commission (NSSC) in Korea. KINS recently acquired a license for SAS4A/SASSYS-1 to support the regulatory obligations over the PG-SFR project (above).
- KTH ELECTRA LFR Concept: The Royal Institute of Technology (Kungliga Tekniska Högskolan) in Stockholm Sweden recently acquired a license for SAS4A/SASSYS-1 to perform natural circulation design performance studies of their ELECTRA lead-cooled fast reactor concept.

In the sections that follow, a brief background is provided for SAS4A/SASSYS-1, and a summary of significant updates that have been released as Version 5.0 is given. The modernization updates that were carried out during FY13 are then described in detail. Documentation updates are relevant to Version 5.0, but the code modernization updates will be released in the near future as Version 5.1. Ongoing updates that will continue into FY14 are also described.

2. SAS4A/SASSYS-1 Developments

2.1 Background

In the late 1960s, the then U.S. Atomic Energy Commission gave development of a liquid-metal-cooled fast reactor (LMR) a high priority, and the development of the Fast Flux Test Facility (FFTF) became a cornerstone of that program. To provide adequate support for the FFTF and for the expected LMRs to follow, a major base technology program was established which provided a continuous stream of experimental information and design correlations. This experimental data would either confirm design choices or prove the need for design modifications. At the time, the "tremendous amount of data and experience pertaining to thermal design" of LMRs was recognized as providing the technical foundation for the future commercial development of LMRs.[7]

Along with the generation of experimental data came the development of safety analysis methods that used that data in correlations for mechanistic, probabilistic, or phenomenological models. These models were developed for a variety of needs ranging from individual components, such as heat exchangers, pumps, or containment barriers, to whole core or even whole-plant dynamics. A major portion of the overall technical effort since that time has been allocated to safety considerations, and the SAS4A/SASSYS-1 safety analysis code is the result of that dedication.

Development of the SAS series of codes[8–13] began in the mid 1960s. SAS has the capability to model the transient behavior of several representative channels to evaluate the initiating phase of HCDAs. SAS1A originated from a sodium-boiling model and includes single- and two-phase coolant flow dynamics, fuel and cladding thermal expansion and deformation, molten fuel dynamics, and a point kinetics model with reactivity feedback.[8] By 1974, SAS evolved to the SAS2A computer code[9] which included a detailed multiple slug and bubble coolant boiling model which greatly enhanced the ability to simulate the initiating phases of loss-of-flow (LOF) and transient overpower (TOP) accidents up to the point of cladding failure and fuel and cladding melting.

The SAS3A code[10] added mechanistic models of fuel and cladding melting and relocation. This version of the code was used extensively for analysis of accidents in the licensing of FFTF. In anticipation of LOF and TOP analysis requirements for licensing of the Clinch River Breeder Reactor Plant (CRBRP), new fuel element deformation, disruption, and material relocation models were written for the SAS4A version of the code,[11] which saw extensive validation against TREAT M-Series test data. In addition, a variant of SAS4A, named SASSYS-1,[12] was developed with the capability to model ex-reactor coolant systems to permit the analysis of accident sequences involving or initiated by loss of heat removal or other coolant system events. This allows the simulation of whole-plant dynamics feedback for both shutdown and off-normal conditions, which have been validated against EBR-II Shutdown Heat Removal Test (SHRT) data and data from the FFTF LOF tests.

Although SAS4A and SASSYS-1 are generally portrayed as two computer codes, they have always shared common code architectures, the same data management strategy, and the same core channel representation. Subsequently, the two code branches were merged into a single code referred to as SAS4A/SASSYS-1. Version 2.1 of the SAS4A/SASSYS-1 code

[11, 12] was distributed to Germany, France, and Japan in the late 1980s, and it serves as a common tool for international oxide fuel model developments.

Beyond the release of SAS4A/SASSYS-1 v 2.1, revisions to SAS4A/SASSYS-1 continued throughout the Integral Fast Reactor (IFR) program between 1984 and 1994,[14] culminating with the completion of SAS4A/SASSYS-1 v 3.0 in 1994.[15, 16] During this time, the modeling emphasis shifted towards metallic fuel and accident prevention by means of inherent safety mechanisms. This resulted in 1) addition of new models and modification of existing models to treat metallic fuel, its properties, behavior, and accident phenomena, and 2) addition and validation of new capabilities for calculating whole-plant design basis transients, with emphasis on the EBR-II reactor and plant,[17] the IFR prototype. The whole-plant dynamics capability of the SASSYS-1 component plays a vital role in predicting passive safety feedback. Without it, meaningful boundary conditions for the core channel models are not available, and accident progression is not reliably predicted.

By the mid 1990s, SAS4A/SASSYS-1 v 3.1 had been completed as a significant maintenance update, but it was not released until 2012.[13]

2.2 Updates to Version 5

In the time since the development of Version 3, several modeling additions and enhancements have been made to meet U.S. Department of Energy programmatic needs. Significant among these are

- Detailed sub-channel models for whole-core analyses to resolve intra-assembly temperature and flow distributions [18]
- 3D visualization capabilities for sub-channel results.
- Extended decay-heat models to support long-term transients and complex, actinide-bearing fuels.
- Support for coupling with external CFD simulations to resolve flow distribution and thermal stratification effects.
- Treatment of axial expansion feedback from assembly duct walls
- Support for spatial kinetics (requires DIF3D-K)
- Extension of the control-system model to include sinusoidal functions that can be used to represent seismic oscillation effects.
- Addition for heavy liquid-metal coolants (lead and lead-bismuth eutectic)
- Support for user-defined coolant properties
- Detailed steam-generator model updates
- Several bug fixes and other enhancements

In addition to the above, a major restructuring of the code has been completed to adapt all source files to free-form source format so that new model developments can be implemented using modern object-oriented practices. This collection of updates was released on FY12 as SAS4A/SASSYS-1 Version 5.0.

3. Modernization Updates

Three key areas were the focus of FY13 modernization activities. These include 1) preparing sub-channel preprocessor and visualization documentation, 2) partial reconstruction of the data management strategy in SAS4A/SASSYS-1 to facilitate future code development and maintenance activities, and 3) adding comprehensive documentation to the SAS4A/SASSYS-1 manual for the updated capabilities described in the preceding section.

3.1 Sub-Channel Preprocessor and Visualization

In a normal SAS4A/SASSYS-1 simulation, a single-pin "channel" model is used to represent the average fuel pin and coolant behavior in single assembly or in a group of assemblies. Multiple channel models are then used to represent the entire core. In a "sub-channel" model, channel models are coupled together to represent the collection of individual coolant channels in a single assembly, thus providing a higher fidelity simulation of the fuel, clad, and coolant temperatures.

The sub-channel model in SAS4A/SASSYS-1 was developed under an International Nuclear Energy Research Initiative (INERI) collaboration between Argonne National Laboratory and the Korean Atomic Energy Research Institute (KAERI) between 2001 and 2005. Although the modeling activities were successfully completed under the INERI collaboration, input data preparation and output data visualization capabilities had not been fully developed. User documentation for the sub-channel model only existed as INERI annual reports and other internal documentation.

Because of the large amount of input data needed to describe a sub-channel model (even for a single assembly) an input processor had been developed to simplify the task. The input processor takes basic assembly geometry information, along with specification of certain options such as friction factor coefficients and heat transfer parameters, and it generates the input data needed to describe sub-channel coupling, flow areas, cross-flow coefficients, and cross-pin conduction coefficients. Manually creating this input is impractical even for small assemblies. Unfortunately, no documentation for the sub-channel input processor had existed prior to this work.

A third problem with the sub-channel model is that it was difficult to verify the integrity of the input. Coupled with this is the difficulty in analyzing the large amount of output generated by a transient simulation. Under the Nuclear Energy Advanced Modeling and Simulation (NEAMS) program, the visualization tool VisIt was being applied to large-scale data analysis problems such as this. Researchers at Lawrence Livermore National Laboratory developed a plug-in for VisIt that allows it to read in the sub-channel data from SAS4A/SASSYS-1 simulations. Although the visualization capabilities for the sub-channel model in SAS4A/SASSYS-1 were well established by the NEAMS program, documentation of the new capabilities was not integrated into the SAS4A/SASSYS-1 manual.

In FY13, considerable effort was dedicated toward establishing documentation for the sub-channel model in three areas:

- Sub-Channel Physics: Documentation from the INERI collaboration between Argonne and KAERI was consolidated as a new section for the SAS4A/SASSYS-1 manual. The documentation describes the formulation of the sub-channel model and the numerical methods of solution.

- Input Processor: No documentation had previously existed for the input processor. Original documentation was constructed based on a careful review of the source code for the processor. Thorough in-line comments were available that provided details of required user input, but few comments were available for the steady-state solver that determined sub-channel flow distributions, cross-flow coefficients, and other parameters that are generated by the code.
- Data Visualization: Internal memos were retrieved that described the required input for the input processor and for SAS4A/SASSYS-1 that are needed to generate the geometry and data visualization files read by VisIt. This information was used to write an appendix to the SAS4A/SASSYS-1 manual.

An important aspect of the input processor and output data visualization capabilities is that they work together. The input processor generates the visualization mesh that represents the sub-channel model in three-dimensional space, while SAS4A/SASSYS-1 generates the temperature data that is projected onto the visualization mesh. Even a small number of sub-channel assemblies can produce tens of gigabytes of data during a typical transient.

The separation of mesh and data components allows flexibility in constraining the amount of data that is generated. Figure 1 shows the different levels of detail that are available as options in the generation of the visualization mesh (only a single axial plane is shown). The level of detail varies from representing the coolant only (LDETL = 1) to representing the coolant, structure, cladding, and detailed fuel structure (LDETL = 4). Likewise, during a SAS4A/SASSYS-1 simulation, the output generated can be limited to coolant temperatures only; coolant and structure temperatures; coolant, structure, cladding, and fuel average temperatures; or all of the preceding *plus* detailed axial and radial temperature profiles for the fuel pin.

The sub-channel model and data visualization capabilities are important components in the International Atomic Energy Agency Coordinated Research Project on the EBR-II Shutdown Heat Removal Tests (SHRT-17 and SHRT-45R). Argonne will use the analysis capabilities to predict sub-channel coolant temperatures in the EBR-II instrumented assemblies XX09 and XX10. In return, thermocouple data from the instrumented assemblies can be used to validate the sub-channel model. Preliminary results for XX09 during the SHRT-17 test are shown in Figure 2. The six assemblies surrounding XX09 are also represented in the model because they are important contributors to the temperature distribution within XX09.

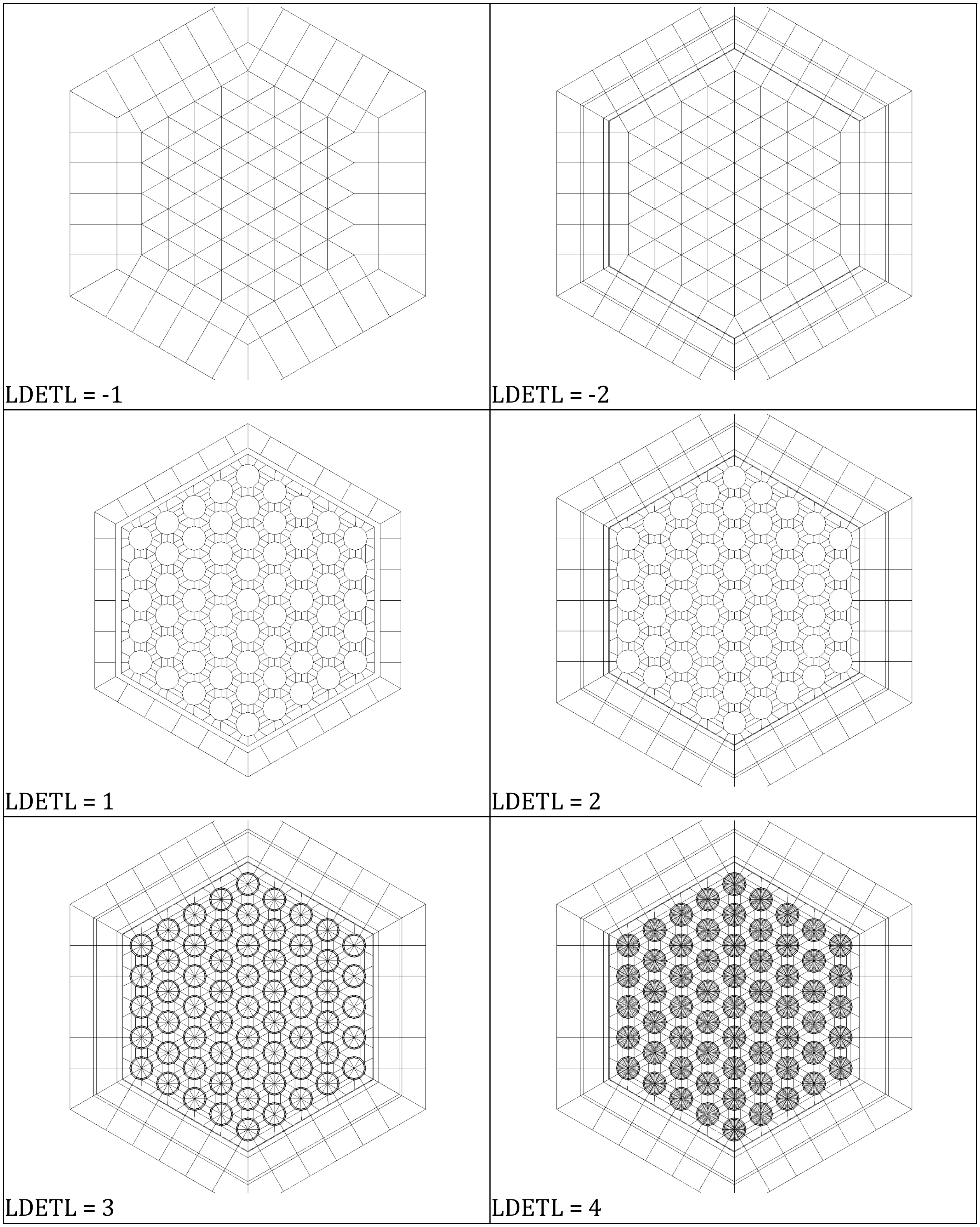


Figure 1: Examples of Geometry Detail Options Applied to the EBR-II XX09 Instrumented Test Assembly.

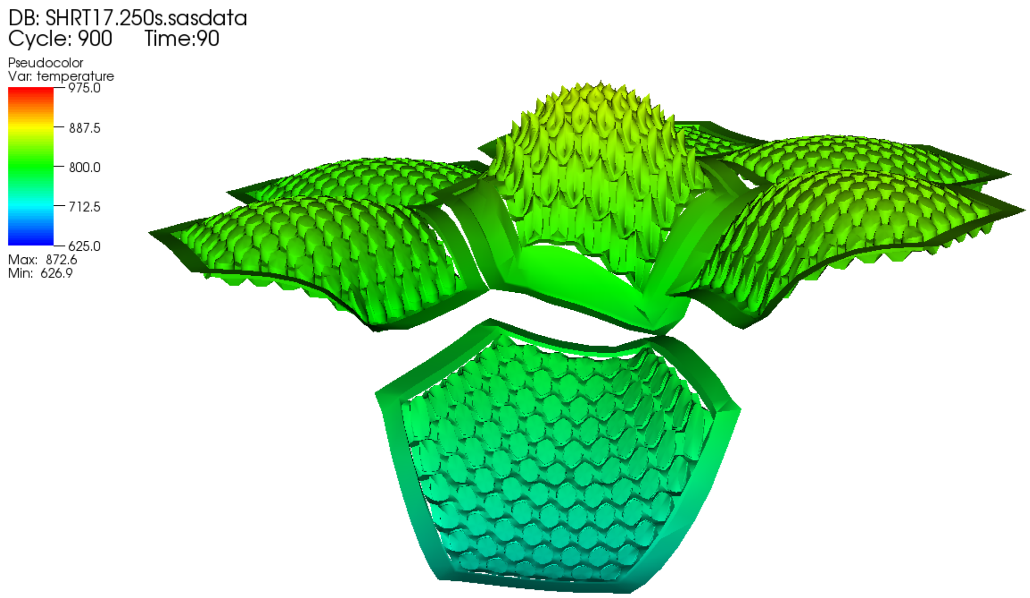


Figure 2: Contour Plot of Predicted Coolant, Structure, and Fuel Temperatures (°K) for XX09 and Adjacent Assemblies during the SHRT-17 Test. (Top of Core at *t* = 90 seconds)

3.2 Data Management

Most of SAS4A/SASSYS-1 originated on computing architectures with extremely limited memory capacities compared to current hardware. Code development practices revolved around minimizing memory usage. A fundamental strategy was to overlay, or reuse, the same memory locations for multiple different models. Data that was not needed for an immediate calculation was cached to some larger capacity storage, such as disk or tape. When part of a calculation was completed, its results would be cached, and other data would be loaded into memory to continue the next phase of the calculation.

In SAS4A/SASSYS-1, where individual core channels represented one or more assemblies in a core design, the natural division of data was at the core channel level. Data for every channel was stored on a large-capacity device. A loop over all channels would first load the data for a single channel, perform the calculations using a shared memory location, and then store the results for that channel before moving on to the next channel.

By the early 1990s, affordable memory capacities had grown to several tens or even hundreds of megabytes. This allowed the memory to act as a large-capacity device, which eliminated the time-consuming process of reading and writing data on disk in favor of the faster process of reading and writing data in memory. A simplified depiction of this process is shown in Figure 3. As shown, Channel 2 is loaded into the shared memory location (blank common) and the simulations for that channel (thermal-hydraulic, fuel performance, and reactivity feedback models) are being computed.

Code developers interpret the shared memory in a certain way in order to deduce the location of simulation variables. In Figure 3, for example, a number of “data packs” are defined, within which are thousands of variables. In the COMC data pack, a specific location is defined for TCBAR1 and TCBAR2, which represent the axial temperature distribution of the coolant at the beginning (1) and end (2) of the current time step. Once completed, the results for Channel 2 will be written back to the large memory location, and Channel 3 will be loaded. This was the predominant data management strategy in SAS4A/SASSYS-1 Version 3.1.

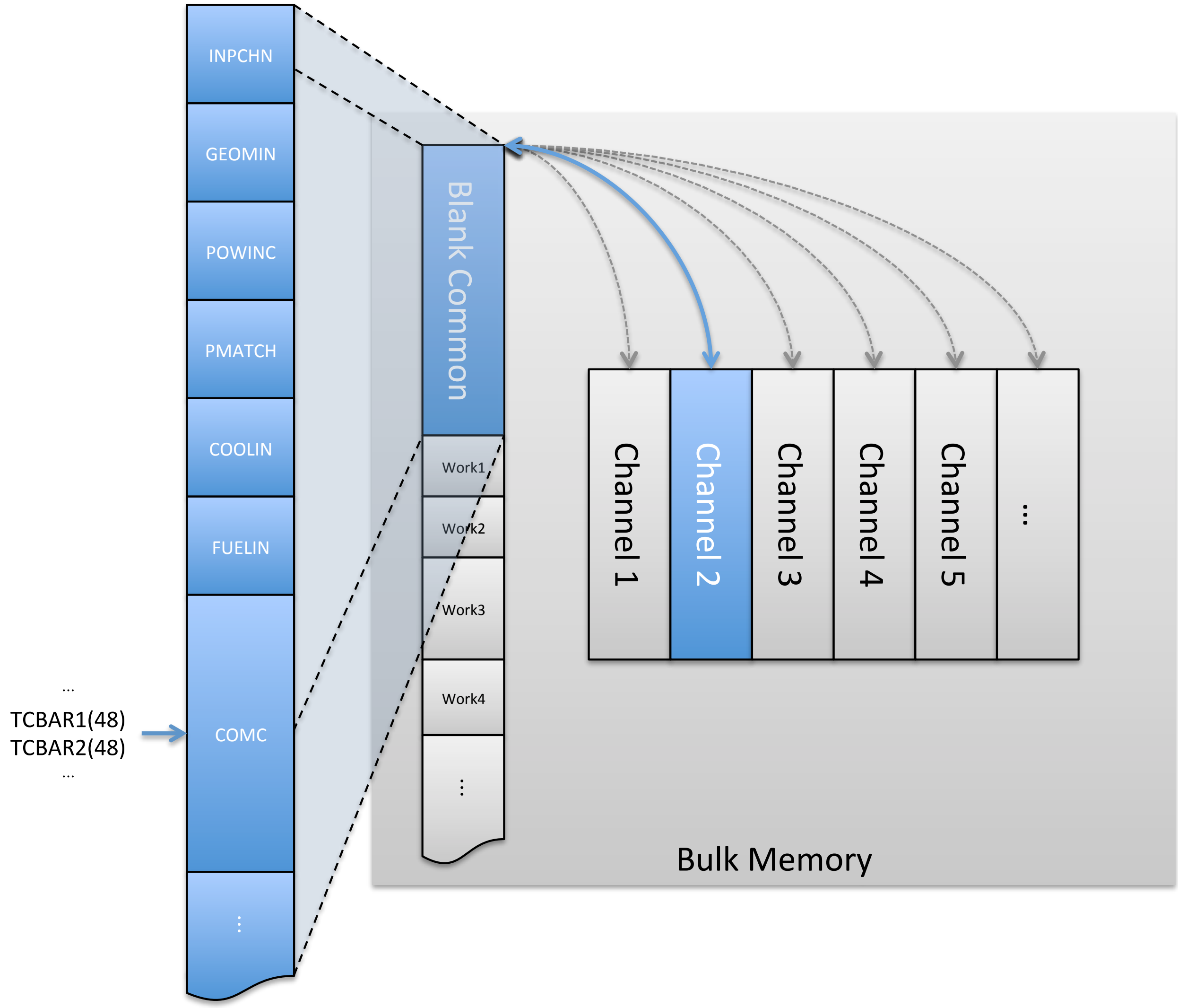
Even though it came with serious drawbacks, this data management strategy was essential for small memory machines. Consumer computing hardware today has three orders of magnitude more memory than in the 1990s, so now the drawbacks impose limitations on code development and performance with no return benefit. These limitations include

- Maintainability: Code improvements that change the length or interpretation of the shared memory location can have unintended, far-reaching consequences. Any data following the change can shift to different memory locations. This affects all models, even unrelated ones, that expect the shared memory to have a fixed interpretation.
- Scale: In theory, large-scale problems could be simulated with SAS4A/SASSYS-1. The current data management strategy is limited to 32-bit addressing, which imposes a practical upper limit of around 10,000 channels. This means full-core sub-channel analysis is not yet possible.
- Speed: Even though data transfer in memory is much faster than data transfer to disk, it is still much slower than direct access. This is especially a problem when severe accident conditions are modeled, because small time steps require more frequent data transfer that severely impairs code performance.
- Parallelism: Because channels are generally independent, they could be solved in parallel. The current channel model depends on a shared memory location, so simultaneous use is not possible, and multi-threaded parallelism is not practical.

To remove these limitations, a number of significant changes are being implemented in the data management strategy in SAS4A/SASSYS-1. Updates that improve code maintainability are being prioritized ahead of scale, speed, and parallelism. To this end, updates have focused on developing object-oriented data modules that encapsulate many of the abstract data packs described above into concrete user-defined data types with well-defined structures. New or existing features of SAS4A/SASSYS-1 that use the new data modules can alter them without concern for disrupting other features of the code because each data module is independent of the others.

This seemingly simple approach required significant effort to develop, implement, and verify. As described in the list above, changes to the shared memory structure disrupts the entire code. Relocating one variable at a time was rejected as being too time consuming since there are thousands of variables that occupy the shared memory of blank common. Instead, entire data pack were moved starting with INPCHN.

When the top data pack is removed, all other data in the shared memory shifts up and the second data pack in the former arrangement now becomes the first. This immediately breaks functionality because many parts of the code rely on the specific layout of the shared memory to locate variables. Furthermore, it changes the *size* of the shared memory, so the layout of the channels in bulk memory also changes.



Channel-dependent data is repeatedly copied into and out of the shared, global "blank common" as a large array. It is up to code developers to ensure a consistent interpretation of the shared memory in order to locate variables, such as TCBAR1 and TCBAR2.

Figure 3: Simplified Overview of Previous Channel-Dependent Data Management in SAS4A/SASSYS-1.

Updating SAS4A/SASSYS-1 to recognize the new layout and size of the shared memory is a tedious, but not difficult, process for the variables that *remain* in the shared memory location. For the variables that are part of the data pack that was *removed*, updating the code to use the new data module was much more challenging. The difficulty of this part of the update is made complex by the fact that each variable that is moved may be accessed in one of two ways:

- In Shared Memory: This is the traditional approach described above. Data is copied from bulk storage into the shared memory, accessed and/or updated by a model, and then written back to bulk storage. For example, the code TCBAR2(1) = 350.0 +

273.15 would assign a value to the inlet temperature of the current channel, but it wouldn't actually get stored until the data transfer step writes it back to bulk storage.

- Directly, by Violating Array Bounds: Data in bulk storage can be read and/or written directly if one knows the memory offset between the shared memory and each channel in bulk storage. For example, TCBAR2(1+IOFFSK) = 350.0 + 273.15 would assign a value to some random location in memory. If IOFFSK = JPTV(ICH) is the memory offset between shared memory and the data for channel ICH in bulk memory, then the value is stored in the correct memory location without the need for data transfer.

It is essential to eliminate both of these access methods because the first requires expensive data transfer at every time step and the second relies on a jump into extended memory locations that is only correct if the code developer maintains an accurate knowledge of all the data locations. Eliminating the second approach had a higher priority in FY13 because it has a detrimental impact on code maintainability.

New data modules have been implemented for seven of the major data packs: INPCHN, GEOMIN, POWINC, PMATCH, COOLIN, FUELIN, and COMC. A hybrid data management approach was devised to significantly reduce the extent of code changes that were needed to use the new modules. Each of the new modules maintains its own, semi-private shared memory to facilitate portions of the existing code that expect to access data via shared memory. However the data from these seven data packs is no longer in blank common. Furthermore, the new data modules maintain their own restart functionality to save and restore data when a calculation is interrupted and then restarted at a later time.

For the portions of the code that relied on array-bounds violations to access simulation data directly, extensive code updates had to be implemented to correct the usage for all variables affected. In the case of TCBAR2, for example, the usage changed from TCBAR2(1+IOFFSK) to CHAN(ICH)%TCBAR2(1). This change eliminates the error-prone tasks of maintaining auxiliary variables (such as IOFFSK, IPTV, JPTV, and others) to keep track of memory management and variable locations. Instead, the compiler constructs a data structure (CHAN) and keeps track of its internal layout. Code developers do not need to know what the layout is, only that the variables are available for use in the simulation.

The hybrid data management approach implemented in FY13 greatly accelerated deployment of the new data modules. But it doesn't completely eliminate the overhead of shared memory data transfers since each module implements a private copy. Nevertheless, the approach allows future updates to be added at an incremental pace without disrupting the entire code.

Although seven of the major data packs have been updated, the data packs that correspond to severe accident phenomena (sodium boiling, in-pin fuel melting and relocation, and ex-pin fuel expulsion) have not been updated.

3.3 Code Documentation

As described in Section 3.1, extensive documentation on the sub-channel modeling capabilities of SAS4A/SASSYS-1 have been prepared. Main documentation of the sub-channel model will appear as a new section in Chapter 3 of the code manual, and the input processor and visualization capabilities are described in an appendix to the same chapter. In addition to these major updates, a number of other significant documentation updates have also been completed. These include

- Updated flow charts for the top-level functions of the code: input processing, steady-state initialization, and transient state solutions (Chapter 2). These flow charts provide considerable detail on the operation of the three driver routines that manage overall program flow.
- A rewritten section describing the updated decay heat model that was developed under the Advanced Fuel Cycle Initiative program (Chapter 4). The updated model provides higher fidelity capabilities for characterizing decay heat from actinide bearing fuels and includes ANS standard decay heat curves for U-235, U-238, Pu-239, and Pu-241.
- Reconstruction of two chapters that were previously only available from scanned paper copies. DEFORM-4 (Chapter 8) describes the detailed pre-transient and transient oxide fuel behavior models, and DEFORM-5 (Chapter 9) describes the metallic fuel and cladding transient failure models.
- A new section on duct wall axial expansion (Chapter 4). TerraPower, LLC funded development of this model under a Work for Others contract, but documentation had not been integrated into the manual.
- Updates to the User's Guide (Appendix 2.2), which provides a comprehensive summary of all user input. This is one of the largest and most-consulted sections of the manual, and input descriptions were added for the new sub-channel, decay heat, visualization, and axial expansion models.

The SAS4A/SASSYS-1 code manual consists of over 2100 pages of detailed documentation on the physics, models, and solution methods used in whole-plant transient simulations. As code revisions take place and additional capabilities are added to SAS4A/SASSYS-1, it will be essential that the manual is kept up to date.

Although the manual will continue to be the primary delivery mechanism for code documentation to both users and developers, a second, automated documentation generation capability has been implemented based on Doxygen. Doxygen is a tool for generating web-based code documentation from annotated source code files. Although very few annotations have been written into the SAS4A/SASSYS-1 source files, Doxygen also generates a detailed web site that describes the source code hierarchy, subroutine dependencies, and call chain maps. This is some of the same information contained in Chapter 2 of the code manual. Manually updating Chapter 2 as code changes are implemented is not practical, and online Doxygen documentation will eventually replace this portion of the manual.

4. Continuing Work

Although revisions to the code manual have been completed, the final review and publication clearance process is still in progress. Part of the challenge is in dealing with such a large document, however the process should be completed by mid October 2013. The citation will be as follows:

T. H. Fanning, ed., *The SAS4A/SASSYS-1 Safety Analysis Code System, Version 5*, ANL/NE-16/19, Nuclear Engineering Division, Argonne National Laboratory.

Once the review is completed, the full documentation will be available through OSTI.

Earlier in this document, it was noted that code modernization updates will be released as SAS4A/SASSYS-1 Version 5.1. Many of the data blocks have been moved out of the shared memory in blank common. The most recent restructuring was for COMC. Nearly all verification test cases complete successfully using the new modules for COMC. The single exception is for a sub-channel test case, which fails to complete beyond three time steps in the simulation. Once this issue is resolved, Version 5.1 will be released.

Some data blocks have not yet been moved out of the shared blank common memory storage. Primary among these are the variables that represent simulation data for severe accidents, such as sodium boiling and fuel melting and relocation. Implementing new data structures for these variables will be a significant focus during FY14.

Other work that is planned includes updates to the plant control system models, modularizing the balance of plant interface to support other power conversion systems (e.g. super-critical CO_2), and creation of an interface to support frequent simulation restarts to support dynamic PRA analyses. Ongoing maintenance of documentation is also required to keep pace with code updates.

5. Summary

Modernization of the data management in the SAS4A/SASSYS-1 source code and updates to the related documentation were completed in FY13 to ensure that SAS4A/SASSYS-1 remains a viable simulation tool for the safety analysis of advanced, low-pressure, non-LWR reactor concepts. Code updates focused on eliminating the legacy data-management practices that dominated code development efforts into the 1980s, before the availability of object-oriented software development practices. A large part of the data management has been reconstructed in a modular way, but some portions — particularly the data for severe accident modeling — still need to be updated.

Code updates that had been completed and released as SAS4A/SASSYS-1 Version 5.0 have now been fully documented in the extensive code manual, which will be released in mid October. One of the most significant omissions was documentation for the sub-channel model, input preprocessor, and data visualization capabilities. Documentation for these has been incorporated into the manual.

6. References

1. R. Schmidt, et al., *Sodium Fast Reactor Gaps Analysis of Computer Codes and Models for Accident Analysis and Reactor Safety*, SAND2011-4145, Sandia National Laboratories, June 2011.
2. T. Sofu and L. Briggs, “Benchmark Specifications for EBR-II Shutdown Heat Removal Tests,” *Proc. ICAPP ’12*, Chicago, IL, June 24-28, 2012.

3. T. Sumner, T. H. Fanning, and T. Sofu, *unpublished information*, Argonne National Laboratory, March 2013.
4. T. Sumner, T.K. Kim, and C. Grandy, *unpublished information*, Argonne National Laboratory, September 2013.
5. D. Grabaskas, T. Sofu, "Review of Passive System Reliability Modeling Approaches for Advanced Small Modular Reactors," *Proc. of the Int'l Topical Meeting on Probabilistic Safety Assessment and Analysis (PSA 2013)*, Columbia, SC, September 2013.
6. A. Hakobyan, T. Aldemir, R. Denning, S. Dunagan, D. Kunsman, B. Rutt, U. Catalyurek, "Dynamic Generation of Accident Progression Event Trees," *Nuclear Engineering and Design*, vol. 238, pp. 3457-3467, 2008.
7. Y. S. Tang, R. D. Coffield, Jr., R. A. Markley, *Thermal Analysis of Liquid Metal Fast Breeder Reactors*, Westinghouse Electric Corporation, Madison, Pennsylvania, American Nuclear Society Publications, 1978.
8. J. C. Carter, et al., *SAS1A, A Computer Code for the Analysis of Fast-Reactor Power and Flow Transients*, ANL-7607, Argonne National Laboratory, October 1970.
9. F. E. Dunn, et al., *The SAS2A LMFBR Accident-Analysis Computer Code*, ANL-8138, Argonne National Laboratory, October 1974.
10. M. G. Stevenson, W. R. Bohl, F. E. Dunn, T. J. Heames, G. Hoppner, and L. L. Smith, "Current Status and Experimental Basis of the SAS LMFBR Accident Analysis Code System," *Proc. Fast Reactor Safety Meeting*, American Nuclear Society, CONF-740401-P3, pp. 1303–3121, Beverly Hills, CA, April 2–4, 1974.
11. G. Birgersson, et al., *The SAS4A LMFBR Accident Analysis Code System*, ANL/RAS 83-38, Revision 2, Volumes 1 and 2, Argonne National Laboratory, Argonne, Illinois, February, 1988.
12. F. E. Dunn, et al., *The SASSYS-1 LMFBR Systems Analysis Code*, ANL/RAS 84-14, Revision 1, Argonne National Laboratory, Argonne, Illinois, March, 1987.
13. T. H. Fanning, ed., *The SAS4A/SASSYS-1 Safety Analysis Code System*, ANL/NE-12/4, Nuclear Engineering Division, Argonne National Laboratory, January 31, 2012.
14. C. E. Till, "Fast Reactor Directions: The LMR Integral Fast Reactor Program at Argonne," *Proceedings of the International Fast Reactor Safety Meeting*, Snowbird, Utah, August 12-16, 1990.
15. J. E. Cahalan, A. M. Tentner, and E. E. Morris, "Advanced LMR Safety Analysis Capabilities in the SASSYS-1 and SAS4A Computer Codes," *Proceedings of the International Topical Meeting on Advanced Reactors Safety*, American Nuclear Society, Vol. 2, pp. 1038–1045, Pittsburgh, PA, April 17–21, 1994.
16. *The SAS4A/SASSYS-1 LMR Analysis Code System*, Volumes 1–5, ANL-FRA-1996-3, Argonne National Laboratory, August 1996.
17. J. Sackett, et al., "EBR-II Test Programs," *Proceedings of the International Fast Reactor Safety Meeting*, Snowbird, Utah, August 12-16, 1990.
18. F. E. Dunn, J. E. Cahalan, D. Hahn, H. Y. Jeong, "Whole Core Sub-Channel Analysis Verification with EBR-II SHRT-17 Test," *Proc. 2006 Int'l Congress on Advances in Nuclear Power Plants*, Reno, NV, June 6–8, 2006.

Nuclear Engineering Division
Argonne National Laboratory
9700 South Cass Avenue, Bldg. 208
Argonne, IL 60439-4842

www.anl.gov

Argonne National Laboratory is a U.S. Department of Energy
laboratory managed by UChicago Argonne, LLC